UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/718,301	11/20/2003	Mazen Faraj	CA920030063US1	9790

46073 7590 02/06/2006

IBM CORPORATION (VE)
C/O VOLEL EMILE
P. O. BOX 202170
AUSTIN, TX 78720-2170

EXAMINER	
WOODS, ERIC V	

ART UNIT	PAPER NUMBER
2672	

DATE MAILED: 02/06/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	10/718,301	FARAJ, MAZEN	
	Examiner	Art Unit	
	Eric Woods	2672	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *17 October 2005*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-22* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-22* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *17 October 2005* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☒ All b)☐ Some * c)☐ None of:

1.☒ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

07

DETAILED ACTION

Response to Arguments

Applicant's arguments, see Remarks pages 1-8, filed 17 October 2005, with respect to the rejection(s) of claim(s) 1-22 under various statutes have been fully considered and are found to be partially persuasive.

The rejection of claims 1-22 under 35 USC 103(a) are **NOT** withdrawn. The allegation of common ownership is utterly irrelevant to the determination of the eligibility of a reference, nor is the issue of whether or not that application has issued. Firstly, the reference was published on June 13, 2002 as US PGPub 2002/0073078 A1. This is more than one year and one day before either the US filing data **OR** the foreign priority date, and it is noted that 35 USC 102(b) applies from the earliest US filing date, not the earliest claimed foreign priority date. In any case, the publication is available under 35 USC 102(b), as having been published more than one year prior to the filing of the US application. Therefore, the reference is eligible under both 35 USC 102(a) and 102(b); thusly, 35 USC 103(c) does **NOT** apply in this situation. Neither the submission of affidavits under 37 CFR 1.130 or 1.131 nor the filing of a terminal disclaimer can overcome the statutory bar.

Applicant has not rebutted the statements of examiner with regards to certain elements being well known in the art. Applicant has failed to respond to all of examiner's other points and the taking of Official Notice and has placed all reliance on arguments concerning the eligibility of the Ku reference. Therefore,

all arguments presented after final will constitute new arguments not entered before final rejection. Applicant has no grounds for appeal in this situation.

Please note that applicant did **NOT** challenge the Taking of Official Notice in the first Office Action on pages 8 and 9. Therefore, applicant cannot raise this issue in the Appeal Brief, and applicant has acquiesced to the correctness of all facts therein. Further, the prosecution history estoppel under *Festo* has been created since applicant did not dispute this point.

However, should applicant raise the issue on appeal, examiner will provide additional evidence to prove that Windows Explorer had all of the recited capabilities as of the date of the reference. This would be provided if the Board so requires, and would not constitute new grounds of rejection or new evidence per se under the guidelines set forth by the Board.

BPAI: **Please note that all arguments presented in the Appeal Brief will be newly introduced and were not presented before final rejection. Applicant has not presented good and sufficient reasons why such arguments were not early presented. As such, they should not be considered as part of the record. There are no grounds for appeal.**

The rejections of claims 18 and 19 under 35 USC 112, first paragraph, stand withdrawn in view of applicant's amendments.

The rejection of claims 2-10, 12-20, and 22 under 35 USC 112, second paragraph, stand withdrawn in view of applicant's amendments.

The objections to the specification stand withdrawn in view of applicant's amendments.

The objections to the drawings stand withdrawn in view of applicant's amendments.

The newly added material to the specification is not new matter under 35 USC 112, first paragraph, since as applicant notes it has prior support in the claims.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

The factual inquiries set forth in *Graham* **v.** *John Deere Co.*, 383 U.S. 1, 148 USPQ 459 (1966), that are applied for establishing a background for determining obviousness under 35 U.S.C. 103(a) are summarized as follows:

1. Determining the scope and contents of the prior art.
2. Ascertaining the differences between the prior art and the claims at issue.
3. Resolving the level of ordinary skill in the pertinent art.
4. Considering objective evidence present in the application indicating obviousness or nonobviousness.

Claims 1-2, 4-11, and 19-22 are rejected under 35 U.S.C. 103(a) as being unpatentable over Windows™ Explorer ('Explorer') and Ku et al (US PGPub 2002/0073078 A1, eligible under 35 U.S.C. 102(b))('Ku') in view of Pajak et al (US 5,388,196).

Firstly, in response to applicant's arguments, the recitation "in a distributed data processing system" has not been given patentable weight because the

recitation occurs in the preamble. A preamble is generally not accorded any patentable weight where it merely recites the purpose of a process or the intended use of a structure, and where the body of the claim does not depend on the preamble for completeness but, instead, the process steps or structural limitations are able to stand alone. See *In re Hirao*, 535 F.2d 67, 190 USPQ 15 (CCPA 1976) and *Kropa v. Robie*, 187 F.2d 150, 152, 88 USPQ 478, 481 (CCPA 1951). Further, the majority of the preamble, "A method of displaying and interacting with local and remote data objects" is clearly merely a summary and possibly an intended use.

Secondly, distributed data processing systems are well known in the art. Any system that accesses files on remote systems (e.g. a web browser) and allows the user to access operations (e.g. web-based email) are prima facie distributed. For example, the American Heritage College Dictionary defines the term "distributed" to mean, "to divide and dispense in portions". Thusly, in the recited context of the claim the term is meaningless.

Further, it is well known that Microsoft® Windows™ has for many years had a feature called "Explorer" or more recently "Windows™ Explorer" that has evolved into a general browser-like interface. In any case, this feature has been present in Windows™ since version 3.1 over a decade ago. This interface allows the browsing of both local objects and remote objects, which can for example be mapped as network drives, NAS / NFS / SMB shares, "My Network Places", and the like in Windows™ Explorer, including the advanced versions which are available under Windows™ 2000 and Windows™ XP. Thusly, that embodiment

is well-known prior art. Examiner takes Official Notice of all of the above; if applicant wishes to challenge it, references will be provided. Applicant is reminded that a challenge or response on this point **must** be made in reply to this Office Action, or the opportunity to use this argument on appeal or in further proceedings will be lost, and will result in the creation of a prosecution history estoppel (see for example *Festo Corp.* vs. *Shoketsu Kinzoku Kogyo Kabushiki Co., Ltd.* (Festo III, 2000)).

The embodiments provided by applicant in the specification are proof of the fact that the term "distributed computing environment" is meaningless also, because they refer to a browsing mechanism in an Eclipse™ IDE (specifically operable on IBM® iSeries™). Therefore, this embodiment is exactly what examiner will direct all rebuttals and arguments towards, which will simplify the issues should the case go to appeal.

Lastly, applicant uses the word "model", which the American Heritage College Dictionary defines to mean, "A schematic description of a system..." Also, in common parlance in computer science, the term is generally taken to mean, "A representation of a data structure or schema". Both definitions are complementary; the latter will be used by examiner in the instant case, and will treated as controlling in all further prosecution unless applicant disputes this point.

Practically, the term model will therefore be taken to mean any graphical representation of a file system and its structure, as in Windows™ Explorer for example, or similar GUI-based file managers as are well known in the art.

Therefore, it should be understood that any file manager or operating system that such a hierarchy therefore meets this limitation.

The system and computer readable medium of claims 11 and 20 respectively are the same thing as claim 1 and are rejected in the same manner. The additional means limitation of claims 18 and 19 are treated as a process (not against *In re Donaldson* since applicant's claims are currently not enabling on that count for means plus function since the specification does not set forth the additional limitations to show what the recited means are. The means limitations of claims 18 and 19, upon addition of the flowchart, show that the 'means' are only steps in a computer program or steps on a flowchart, otherwise covered by the computer program and/or method claims generally. The 'system' is merely a computer executing the method or computer program).

As to claims 1, 11, 19, and 21,

A method of displaying and interacting with local and remote data objects in a distributed data processing system, comprising: (see Ku Title and Abstract) (i) Accessing a local model defining at least one local data object and at least one local action, which may be performed on, said local data object; (Ku [0015-0019] for showing model – e.g. local file system / structure, if user is browsing a local repository [0002], then [0046, 0055], where there is the Action menu. Further, see Fig. 4 where each object found in a search has a "repository" location and a "host name" field, both of which indicate local vs. remote location of the object)(Explorer, which when opened prima facie shows a view of the local file system, where all objects shown are "local data objects" and right-clicking on

any object opens a menu that allows the user to edit, delete, rename, or perform a variety of other operations upon any such data object)(Pajak Figs. 3-6 show very clearly a file structure, which again as stated above must be inherent for a local file system, better illustrated in Fig. 7. Specifically, in Fig. 14 it is shown some files are local and some are remote as indicated by the locations shown on the chart)(Note: A file is a type of object. Further Ku teaches the user of objects).

(ii) Accessing a remote model defining at least one remote data object and at least one remote action, which may be performed on, said remote data object; (Explorer as stated above – NFS/NAS shares mapped as a network drive have the same properties to Windows as local files, but those remote data repositories or servers must prima facie have a file system with a hierarchy, which would be accessed by Windows Explorer to create the graphical representation shown in the Windows Explorer window. Also, network drives and locations listed under "My Network Places" would be visible, which again require a remote file system and have the same options. Specifically, right clicking on an object on a remote server allows the same actions to be taken upon the object.)(Ku as above – the same logic for showing local data repositories will also be shown for remote data repositories, please see for example Fig. 4 as set forth in the discussion concerning the previous clause immediately above) (Pajak Figs. 3-6 show very clearly a file structure, which again as stated above must be inherent for a local file system, better illustrated in Fig. 7. Specifically, in Fig. 14 it is shown some

files are local and some are remote as indicated by the locations shown on the chart)

(iii) Displaying at least one of said local data objects and said remote data objects in a viewer; (Ku clearly shows in Fig. 4 the location of objects, and clearly that is a viewer, and it can search across multiple repositories simultaneously.)(Explorer is prima facie a viewer, and it shows the data objects – both remote and local in one window for the user to see, with the hierarchy for navigation in the left hand side of the window.) (Pajak Figs. 3-6 show very clearly a file structure, which again as stated above must be inherent for a local file system, better illustrated in Fig. 7. Specifically, in Fig. 14 it is shown some files are local and some are remote as indicated by the locations shown on the chart. Clearly, objects are shown in the same viewer regardless of their location, but as shown in Fig. 14, objects are very clearly shown with different borders based on where they reside, e.g. as in 16:40-17:15, more details 17:16-19:10 where it is clearly stated the objects that are stored remotely have black borders as shown in Fig. 14)

(iv) In response to selection of a data object from said viewer in (iii), determining a location characteristic for said selected data object; and (Objects in most advanced operating systems have properties inherent to the object, e.g. location. However, in Ku it is very clearly shown where objects are found in a search between different repositories, so that information would be inherently be done by a selection of an object, e.g. the search function would find the object, and the system would prima facie then determine characteristics concerning the

object.)(Pajak provides this step, since Pajak provides that objects in one folder can be located in various places across both local and remote storage, which means that clearly the system would have to determine the properties of an object and its location when selected by the user. Clearly, Pajak determines location as determined by the icon shown to the user as in Fig. 14 and 16:450-18:15. More to the point, in 18:47-19:10, it is clearly taught that the status column indicates several things concerning objects, that the icon will be shown as local with the small terminal (with the outline if remote), whether or not the file has been locked, and whether or not the version on the desktop is more recent that the version on the server. Finally, this would have been obvious, since all three of the references allow the user to select objects in the viewer)

(v) In the context of said location characteristic determined in (iv), performing at least one action on said selected data object as defined by one of said local model and said remote model. (Explorer allows the user to perform operations on both local and remote objects, but in both cases the underlying file system must translate the commands into native actions (e.g. if the NAS / NFS / SMB share is on a server running Linux™, a UNIX™ variant, or other non-Microsoft® operating systems the commands must be translated into file system native commands for the action in question, be it a deletion, a move, a copy, or the like, as are well known in the art)(Context would thusly be inherent in the command under those cases)(Ku clearly teaches that path and location as well as hostname are found for objects in both local and remote repositories for the recited data objects)(Pajak clearly teaches the recited limitation, because as

stated above, operations depend entirely on the context in which they are executed, including permissions (e.g. the user's access to the server may be read-only so that the user can download a copy of the object, but not execute or write to the remote directory – see 8:35-65 for proof this is well known in the art).)

All the limitations are met by all the references as set forth above. Context for operations can be provided by the fact that as Pajak teaches, it is well known in the art that files have permissions and operations may not be valid upon a remote or local file for that reason (e.g. a user may have local write / delete access but only have remote read access as set forth above) so operations will be context dependent. Motivation for combination comes from several different aspects. The Windows™ Explorer shell is well known in the art to provide a general, easy-to-use graphical user interface to examine file systems and files. However, Explorer does not expressly provide for handling objects, whereas Ku does, so as to allow the Explorer interface to be extensible to object repositories and be useful in programming applications and the like. Clearly, this extensibility would provide motivation for combination, along with the object-based searching of Ku. Pajak provides a user interface that clearly illustrates whether or not files are in use (e.g. by the locked status), whether or not they are remote or local (e.g. by the outline), and whether or not they are more recent than the version on the server. This functionality would be exceptionally useful for programmers for the reasons set forth above, and would allow users to determine whether or not a group document or program being collaboratively edited in a group workspace had been updated versus the copy on their workstation. Also, the icons and

status indicators of Pajak would add functionality to Explorer that would give it many more capabilities with respect to allowing the user to determine the status of files and directories at a glance. For the reasons set forth above, it would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the inventions of the various references as set forth above.

As to claims 2, 20, and 22

The method of claim 1, further comprising:

(vi) Defining a local/remote data object comprising a local data object and a remote data object and displaying said local/remote data object in said viewer in (iii);

(vii) If said local/remote data object is selected in (iv), then in (v) performing at least one action on said selected local/remote data object as defined by at least one of said local model and said remote model.

The other two (e.g. Explorer and Ku) references do not expressly teach this limitation.

First of all, clearly as taught in the rejection to claim 1, Pajak teaches that in 16:48-17:3 that when an object is on the Workstation Shared Book and it is locked by the user and opened, a copy will be created locally and will be shown on the desktop as being local, locked (by the user), and more recent than the version on the server. It would be obvious that if a user created a file in the Workstation Shared Book that such a file would have no content and thusly should be locked by the user who created it until a version is ready to be uploaded in order to maintain consistency in the database, which Pajak is very

concerned with (for example, in the Cedar File System, from PARC –which Pajak worked at – various different mechanisms to maintain consistency were used. See attached reference). Since maintaining truth in the Workstation Shared Book is the primary goal of this reference, it would be logical as stated before to have the file locked to its creator until the creation is completed. Obviously, such an object would be shown in the viewer upon creation as explained above.

Next, the step of performing an action is taught as above by Pajak, since the user can obviously perform operations on the local copy of the object, and clearly all three references teach allowing the user to perform operations upon an object. Clearly the second limitation is essentially meaningless, as the user can obviously perform operations upon an object, and it would be obvious that the operations would be limited to those that the user could perform subject to permissions and operations allowed on the native file systems of remote hosts. Motivation and combination is incorporated from the rejection to the parent claim. Clearly, the operations would be performed upon the local object, and when that version was synchronized against / uploaded to the remote database server, clearly the operation would be performed against both objects if allowed. The claim is being interpreted as set forth in the rejection of this claim under 35 U.S.C. 112, second paragraph, as above with respect to *SuperGuide v. DirecTV* (CAFC, 2004, page 20, section IV-B, "At least one of") as requiring both rather than the alternative language.

As to claim 4,

The method of claim 2, further comprising displaying in said viewer a list of local actions and a list of remote actions that may be performed on said local/remote data object.

Clearly, a hybrid data object as listed above that had both existence locally and remotely would allow users to perform certain actions against a local object (e.g. writes, changes, and other alterations). Further, assuming that the user locked the object, the user could then write it back. If however the lock was write-only, e.g. other users could still read the object on the server (which is well known, e.g. allowing the user to have file permissions such that a file is read-only) then the system of Pajak would still create a local version of the object as stated before. However, the user would not have write-back permission until the person using the object finished and released the lock upon it and the user then obtained the lock. This is merely an example as how there could be different local and remote options available for an object. Clearly, since the hybrid object would have been obvious, it would be obvious to expose the command lists that the user can perform on both versions of the object in the viewer, since it spans both locations for the reasons set forth in the rejection to claim 1.

As to claim 5,

The method of claim 2, further comprising displaying in said viewer a list of actions that may be performed on said local/remote data object, and upon selection of an action, further displaying a list of locations on which said action is to be performed.

Obviously in Explorer, right clicking on an object exposes a list of commands or actions that can be performed upon an object. If an object representation exists in more than one place (e.g. the hybrid object), it would be obvious that the user should be able to determine which version(s) to modify, change, or otherwise perform changes to. For example, if a rename command were issued, it would be obvious to let the user choose which location the rename command would be applied to, because the path on the server might be important (for example, say the file was stored in a web directory as /web/foo.html with CGI scripts targeting that location and locally as foo.html; a rename operation on the web server might change the entire structure of the web site and thusly a rename operation would not be wise; therefore, determining a location would be obvious). Although the example provided is somewhat contrived, it is a good, common sense example of how paths on files (and dependencies on file names) can be very important. Again, the existence of a hybrid object necessitates fine-grained control over it for at least the above reasons. Motivation and combination are incorporated by reference from the parent claim

As to claim 6, this would be obvious – as in Pajak, the local version can be synchronized with the remote version when the user has the lock for the file, and obviously as set forth above, the object exists in a local version and a remote version. Therefore, allowing the user to choose where to apply an operation – either to A or B or to the combination of both – would be obvious in view of the parent claim for the reasons set forth in the rejection therein.

As to claim 7, this limitation is met in the rejection to claims 5 and 6 above – if an action is applied to both objects, it would be obvious to apply it simultaneously to both objects in order to preserve concurrency – if the user had the lock for the server object, it would be obvious that if the user desired to apply for example a delete operation that it should be applied to both objects if that user indicates that desire (see for example claim 6, where the user can indicate whether or not the operation is applied to one or both locations. Motivation and combination is taken from claim 2, and the rejection to claims 5 and 6 are incorporated by reference.

As to claim 8, for at least the reasons set forth above in the rejection to claim 7, this limitation is obvious, and that rejection is incorporated by reference.

As to claim 9, this claim turns back to the rejection to claim 1. Pajak clearly teaches the recited limitation, because as stated above, operations depend entirely on the context in which they are executed, including permissions (e.g. the user's access to the server may be read-only so that the user can download a copy of the object, but not execute or write to the remote directory – see 8:35-65 for proof this is well known in the art.) Therefore, it would have been obvious, and the motivation and combination of claim 1 is incorporated by reference.

As to claim 10, this is an obvious variant. If the valid actions are determined, obviously those would be presented to the user upon request based on the user's permissions and the examples provided, for example if the user does not have write permission for the server object, that limitation should not be

presented to the user for the remote object, but could be for the local object (see the discussion of that limitation in previous rejections above.

Claims 3 and 12-18 are rejected as unpatentable under 35 U.S.C. 103(a) over Explorer, Ku, and Pajak as applied to claim 2 above, and further in view of Rich et al (US 6,457,065 B1, eligible under 35 U.S.C. 102(a)).

Note that consistency is a major concern in the art – see Sun et al as an example (Sun, Chengzheng and David Chen. "Consistency Maintenance in Real-Time Collaborative Graphics Editing Systems".)

As to claim 3,

The method of claim 2, further comprising merging said local model and said remote model into a merged viewer model, said merged viewer model containing data objects and actions from both said local model and said remote model.

The main references do not per se teach this limitation, but Explorer utilizes one window to show objects, and in light of Explorer and Ku it would have been obvious to provide a way to merge the two views as recited. That being said, Rich clearly teaches that objects exist within a hierarchy (Figure 5), and that multiple versions of entities can exist (Figure 6), where element 620 has two prior versions, 621 and 622 (13:28-14:67), and it is well known that hierarchical transactions can execute, and in 15:1-45 it is taught that each object in a hierarchy tree has an internal modified flag so that versions can be synchronized internally on the transaction and tree and also has an external synchronization flag to check the version against some persistent data store. Clearly, there is a need – at some point – to synchronize the versions of the objects. This problem

results in many different techniques, such as concurrency locks (see Pajak) but in time the completed version must be uploaded to the persistent storage or remote site and the two versions must be merged or synchronized.

Clearly, the idea of merging versions is common, as in Pajak where 18:48-62 teaches that the user can lock an object, use the "Check in" command to update the "truth" of the database from locally modified files, and then unlock it. Again, it is well known in the art that the merge must eventually happen.

As taught by Pajak and Rich, synchronizing data for merge is well known in the art. All of that having been said however, it would have been obvious to one of ordinary skill in the art at the time that the invention was made that one representation of an object could be used to hold both local and remote commands for all the reasons specified above.

Examiner asserts that it would have been obvious that multiple versions of one file (the local one and remote one, with the local indicated as being indicated local with the tags of Pajak attached to it) could also be shown in one view, where the older version of the one in the database and the more recent local version would be shown together to establish the linkage to the user, and that the user could execute a synchronize command to merge them – to make them the same as explained above. Each object could also have a flag as in Rich to determine what version it is, so that it could easily be overwritten.

Either implementation would satisfy the language of the claim. In any case, motivation for combining Rich comes from the fact that although Pajak teaches that each object knows its attributes – e.g. they are displayed as

graphical icons on the object, the idea of version flags so that the system knows whether it is internally synchronized with all local implementations and another one so that it knows if it is externally synchronized with a remote data store would obviously be helpful, particular if the object were being used in local transactions by the local environment.

As to claim 12, this is a duplicate of claim 3, and the rejection to claims 2 and 3 is incorporated by reference.

As to claim 13, clearly a viewer under Windows Explorer allows the user to select an object with the mouse, as do all modern graphical user interface viewer systems; even Pajak does this limitation. Motivation and combination is taken from the rejection to the parent claim.

As to claim 14, it is clearly a duplicate of claim 4 with different dependency, so that rejection is copied below. Motivation and combination are incorporated form the rejection to the parent claim.

Clearly, a hybrid data object as listed above that had both existence locally and remotely would allow users to perform certain actions against a local object (e.g. writes, changes, and other alterations). Further, assuming that the user locked the object, the user could then write it back. If however the lock was write-only, e.g. other users could still read the object on the server (which is well known, e.g. allowing the user to have file permissions such that a file is read-only) then the system of Pajak would still create a local version of the object as stated before. However, the user would not have write-back permission until the person using the object finished and released the lock upon it and the user then

obtained the lock. This is merely an example as how there could be different local and remote options available for an object. Clearly, since the hybrid object would have been obvious, it would be obvious to expose the command lists that the user can perform on both versions of the object in the viewer, since it spans both locations for the reasons set forth in the rejection to claims 1 and 11.

As to claim 15, this claim is a duplicate of claim 5. The rejection is repeated below. Motivation and combination are incorporated form the rejection to the parent claim.

Obviously in Explorer, right clicking on an object exposes a list of commands or actions that can be performed upon an object. If an object representation exists in more than one place (e.g. the hybrid object), it would be obvious that the user should be able to determine which version(s) to modify, change, or otherwise perform changes to. For example, if a rename command were issued, it would be obvious to let the user choose which location the rename command would be applied to, because the path on the server might be important (for example, say the file was stored in a web directory as /web/foo.html with CGI scripts targeting that location and locally as foo.html; a rename operation on the web server might change the entire structure of the web site and thusly a rename operation would not be wise; therefore, determining a location would be obvious). Although the example provided is somewhat contrived, it is a good, common sense example of how paths on files (and dependencies on file names) can be very important. Again, the existence of a hybrid object necessitates fine-grained control over it for at least the above

reasons. Motivation and combination are incorporated by reference from the parent claim

As to claim 16, this is a duplicate of claim 6, the rejection to which is incorporated by reference. Motivation and combination are incorporated by reference from the rejection of the parent claim.

As to claim 17, this claim turns back to the rejection to claim 1. Pajak clearly teaches the recited limitation, because as stated above, operations depend entirely on the context in which they are executed, including permissions (e.g. the user's access to the server may be read-only so that the user can download a copy of the object, but not execute or write to the remote directory – see 8:35-65 for proof this is well known in the art.) Therefore, it would have been obvious, and the motivation and combination of claims 11 and 12 are incorporated by reference.

As to claim 18, this limitation is clearly a duplicate of claim 10, the rejection to which is incorporated by reference. Obviously, once invalid actions have been determined, it would be obvious to only show valid actions to the user, because the other actions would not make sense to display, as they would be inactive or useless, and thusly would simply occupy valuable screen real estate. Motivation and combination are taken from the rejection to claim 12 above.

Conclusion

THIS ACTION IS MADE FINAL. Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Eric Woods whose telephone number is 571-272-7775. The examiner can normally be reached on M-F 7:30-5:00.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Michael Razavi can be reached on 571-272-7664. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Eric Woods

JEFFERY BRIER
PRIMARY EXAMINER

31 January 2006